

# 分布式 IoTDB 线程整理(2021.1)

## 线程整理

分组	线程名	数量	说明
心跳和选举相关线程	{nodename}-MetaHeartbeatThread@{创建时间戳}	每个节点 1 个	发送心跳、选举、检测 leader 超时
	{nodename}-DataHeartbeatThread@{创建时间戳}	每个 raft 组 1 个，每个节点有副本数个	同上
	{name}-SerialToParallel{序号}	每个 raft 组个数为副本数 - 1 个，每个节点最多有(副本数 - 1) * 副本数个  注：不触发换主时只有 leader 才有此线程，follower 不存在	同步客户端模式下用来将发送选举和心跳 rpc 并行化，仅在同步客户端模式下才存在
	commitLogPool (暂未指定线程名)	每个 raft 组个数为 1 个，每个节点最多有副本数个  注：follower 才有此线程，leader 不存在	follower 用来异步 commit leader 心跳信息的线程
RPC服务端线程	MetaServerThread-	每个节点 1 个	监听 internal_meta_port 响应请求并产生 MetaClientThread
	MetaHeartbeatServerThread-	每个节点 1 个	监听 internal_meta_port + 1 响应请求并产生 MetaHeartbeatClientThread
	DataServerThread	每个节点 1 个	监听 internal_data_port 响应请求并产生 DataClientThread
	DataHeartbeatServerThread-	每个节点 1 个	监听 internal_data_port + 1 响应请求并产生 MetaHeartbeatClientThread
	ClusterClientServer	每个节点 1 个	监听 cluster_rpc_host 响应请求并产生 ClusterClient
RPC客	MetaClientThread- {序号}	每个节点有：	处理meta组成员间的

客户端线程		0 或 1（取决于是否被 leader catch up 过） + metaSendLogParallelism (meta leader 有几个线程在并发地向该节点发送日志，进一步取决于有多少客户端线程在执行 meta 操作以及是否使用 LogDispatcher) + metaForwardNum (同时有多少 meta 请求被转发到该节点)	RPC 请求（不包含心跳）
	MetaHeartbeatClientThread- {序号}	每个节点： leader 有 0 个， follower 有 1 个。	处理 meta 组成员间的心跳请求
	DataClientThread- {序号}	每个 raft 组有： 0 或 1（取决于是否被 leader catch up 过） + dataSendLogParallelism (data leader 有几个线程在并发地向该节点发送日志，进一步取决于有多少客户端线程在执行 data 操作以及是否使用 LogDispatcher) + dataForwardNum (同时有多少 data 请求被转发到该节点)  每个节点的总线程需要考虑属于该节点的副本数个 raft 组的线程总数	处理 data 组成员间的 RPC 请求（不包含心跳）
	DataHeartbeatClientThread- {序号}	每个 raft 组： leader 有 0 个， follower 有 1 个。  每个节点的总线程需要考虑属于该节点的副本数个 raft 组的线程总数	处理 data 组成员间的心跳请求
	ClusterClient {序号}	客户端连接数	处理客户端的 RPC 请求
异步 rpc selector 线程 (仅在异步客户端模式下存在)	asyncHeartbeatClientPool	每个 raft 组： 1 / 3 CPUCoreNum  每个节点的总线程需要考虑属于该节点的副本数个 raft 组的线程总数	发送心跳的 selector 线程
	asyncSendLogClientPool	每个 raft 组： 1 / 3 CPUCoreNum  每个节点的总线程需要考虑属于该节点的副本数个 raft 组的线程总数	只为 appendEntry 用的 selector 线程池

	asyncClientPool	<p>每个 raft 组： 1 / 3 CPUCoreNum</p> <p>每个节点的总线程需要考虑属于该节点的副本数个 raft 组的线程总数</p>	为选举，查询等用的 selector 线程池
定时线程	NodeReportThread	1	每 10 秒将该节点各 raft member 的日志进度以及操作耗时等信息进行打印
	hardLinkCleaner	1	每 1 小时将该节点 snapshot 产生的超过 3 天以上的硬链接删除
日志复制相关线程	{nodename}-CatchUpThread{序号}	<p>每个 raft 组： follower 无此线程 leader 根据需要的 catchup 的个数来决定，最少 0 个，最多副本数 - 1 个。</p> <p>每个节点的总线程需要考虑属于该节点的副本数个 raft 组的线程总数</p>	catchup follower 的线程
	{nodename}-AppendLog{序号}	<p>每个 raft 组： follower 无此线程 小于三副本的 leader 无此线程 三副本以上的 leader 根据客户端连接的并行度来决定，最少为 0 个，最多为 CPUCoreNum * 10 个</p> <p>每个节点的总线程需要考虑属于该节点的副本数个 raft 组的线程总数</p>	用来并发复制日志的线程池
dispatcher 模式独有线程 (仅在使用 LogDispatcher 时存在)	DispatcherEncoder-{序号}	每个进入发送队列并且尚未序列化的日志对应一个，每个节点最多有 CPUCoreNum 个此线程	在 Log 位于发送队列的同时对其进行序列化
	LogDispatcher-{nodename}-{receiver}	<p>每个 raft 组的 leader 有副本数 - 1 个此线程，follower 没有此线程。</p> <p>每个节点的总线程需要考虑属于该节点的副本数个 raft 组的线程总数</p>	批量获取发送队列中的日志向 follower 发送
snapshot 相关线程	pullSnapshotHintService (暂未指定线程名)	每个 raft 组： 0 或 1 (取决于是否向其他节点拉取或 snapshot)	每 1 分钟向拉取 snapshot 的数据源节点更新进度

		每个节点的总线程需要考虑属于该节点的副本数 每个 raft 组的线程总数	
	pullSnapshotService (暂未指定线程名)	每个 raft 组: $0 \leq x \leq \text{CPUCoreNum}$  每个节点的总线程需要考虑属于该节点的副本数 每个 raft 组的线程总数	用来拉取 snapshot, 即底层的 tsfile 以及 schema 等等
raftLog Manager 相关线程	raft-log-delete-{nodename}	每个 raft 组 1 个, 每个节点有副本数个	定期删除日志线程 (内存控制)
	check-log-applier-{nodename}	每个 raft 组 1 个, 每个节点有副本数个	定期检查日志 apply 进度线程, 用来保证 snapshot 的正确性
	persist-log-delete-{dir}	每个 raft 组 1 个, 每个节点有副本数个	定期删除老的持久化日志线程 (磁盘容量控制)
负载相关线程	{nodename}-{sg}	每个存储组对应一个异步 apply 线程	raft 的异步 apply 线程, 做了并行优化。

## 三节点两副本实例

### node1



```

Data(127.0.0.1:9003)-SerialToParallel0
Data(127.0.0.1:9003)-SerialToParallel1
Data(127.0.0.1:9007)-CommitLog0
Data(127.0.0.1:9007)-HeartbeatThread@1611063889313
Data(127.0.0.1:9007)-root_group_0
DataClientThread-1
DataClientThread-10
DataClientThread-11
DataClientThread-12
DataClientThread-13
DataClientThread-14
DataClientThread-15
DataClientThread-16
DataClientThread-17
DataClientThread-18
DataClientThread-19
DataClientThread-2
DataClientThread-20
DataClientThread-3
DataClientThread-4
DataClientThread-5
DataClientThread-6
DataClientThread-7
DataClientThread-8
DataClientThread-9
DataHeartbeatClientThread-1
DataHeartbeatClientThread-2
DataHeartbeatServerThread-
DataServerThread-
Flush-WAL-Thread-0
HardLinkCleaner
MergeTaskCleaner
Meta-CommitLog0
Meta-HeartbeatThread@1611063888844
MetaClientThread-1
MetaClientThread-2
MetaClientThread-3
MetaClientThread-4
MetaHeartbeatClientThread-1
MetaHeartbeatClientThread-2
MetaHeartbeatServerThread-
MetaServerThread-
NodeReportThread
RPC-ServerServiceImpl
TAsyncClientManager#SelectorThread 36
TAsyncClientManager#SelectorThread 40
sun.misc.Unsafe.park(boolean, long)
TAsyncClientManager#SelectorThread 36
TAsyncClientManager#SelectorThread 40
check-log-applier-Meta
check-log-applier-Node(ip:127.0.0.1, metaPort:9003, nodeIdentifier:311748974, dataPort:40010, clientPort:55560)
check-log-applier-Node(ip:127.0.0.1, metaPort:9007, nodeIdentifier:311753989, dataPort:40014, clientPort:55562)
logback-1
logback-2
logback-3
persist-log-delete-sbin/./data/system/raftLog/0/
persist-log-delete-sbin/./data/system/raftLog/311748974/
persist-log-delete-sbin/./data/system/raftLog/311753989/
pool-2-ioTDB-Flush-ServerServiceImpl-thread-1
pool-2-ioTDB-Flush-ServerServiceImpl-thread-2
pool-2-ioTDB-Flush-ServerServiceImpl-thread-3
pool-2-ioTDB-Flush-ServerServiceImpl-thread-4
pool-3-ioTDB-Flush-SubTask-ServerServiceImpl-thread-1
pool-3-ioTDB-Flush-SubTask-ServerServiceImpl-thread-2
pool-32-thread-1
pool-5-ioTDB-Compaction-ServerServiceImpl-thread-1
pool-5-ioTDB-Compaction-ServerServiceImpl-thread-10
pool-5-ioTDB-Compaction-ServerServiceImpl-thread-2
pool-5-ioTDB-Compaction-ServerServiceImpl-thread-3
pool-5-ioTDB-Compaction-ServerServiceImpl-thread-4
pool-5-ioTDB-Compaction-ServerServiceImpl-thread-5
pool-5-ioTDB-Compaction-ServerServiceImpl-thread-6
pool-5-ioTDB-Compaction-ServerServiceImpl-thread-7
pool-5-ioTDB-Compaction-ServerServiceImpl-thread-8
pool-5-ioTDB-Compaction-ServerServiceImpl-thread-9
pool-5-thread-1
pool-6-thread-1
pool-9-ioTDB-open-files-manager-thread-1
timedQuerySqlCountThread
timedQuerySqlCountThread
sun.misc.Unsafe.park(boolean, long)

```

## node2

```

All tthread groups
main
ClusterClientServer
Data(127.0.0.1:9003)-HeartbeatThread@1611063889363
Data(127.0.0.1:9005)-HeartbeatThread@1611063889327
Data(127.0.0.1:9005)-SerialToParallel0
Data(127.0.0.1:9005)-SerialToParallel1
DataHeartbeatClientThread-1
DataHeartbeatClientThread-2
DataHeartbeatServerThread-
DataServerThread-
HardLinkCleaner
MergeTaskCleaner
Meta-CommitLog0
Meta-HeartbeatThread@1611063885785
MetaClientThread-1
MetaClientThread-2
MetaClientThread-3
MetaClientThread-4
MetaHeartbeatClientThread-1
MetaHeartbeatServerThread-
MetaServerThread-
NodeReportThread
RPC-ServerServiceImpl
TAsyncClientManager#SelectorThread 34
TAsyncClientManager#SelectorThread 38
check-log-applier-Meta
check-log-applier-Node(ip:127.0.0.1, metaPort:9003, nodeIdentifier:311748974, dataPort:40010, clientPort:55560)
check-log-applier-Node(ip:127.0.0.1, metaPort:9005, nodeIdentifier:311750563, dataPort:40012, clientPort:55561)
logback-1
logback-2
logback-3
persist-log-delete-sbin/./data/system/raftLog/0/
persist-log-delete-sbin/./data/system/raftLog/311748974/
persist-log-delete-sbin/./data/system/raftLog/311750563/
pool-5-thread-1
pool-6-thread-1
timedQuerySqlCountThread
timedQuerySqlCountThread

```

## node3

